Invited paper

Rule Based Speech Signal Segmentation

Mindaugas Greibus and Laimutis Telksnys

Recognition Processes Department, Institute of Mathematics and Informatics, Vilnius, Lithuania

Abstract—This paper presents the automated speech signal segmentation problem. Segmentation algorithms based on energetic threshold showed good results only in noise-free environments. With higher noise level automatic threshold calculation becomes complicated task. Rule based postprocessing of segments can give more stable results. Off-line, on-line and extrema types of rules are reviewed. An extrema-type segmentation algorithm is proposed. This algorithm is enhanced by a rule base to extract higher energy level segments from noise. This algorithm can work well with energy like features. The experiments were made to compare threshold and rule-based segmentation in different noise types. Also was tested if multi-feature segmentation can improve segmentation results. The extrema rule-based segmentation showed smaller error ratio in different noise types and levels. Proposed algorithm does not require high calculation resources. Such algorithm can be processed by devices with limited computing power.

Keywords—rule base, speech analysis, speech endpoint detection, speech segmentation.

1. Introduction

Speech segmentation is a process of labeling signal areas with symbolic information in some application. Speech segmentation is important to various automated speech processing algorithms: speech recognition, speech corpus collection, speaker verification etc. In many papers speech segmented using wavelet [1], fuzzy methods [2], artificial neural networks [3] and hidden Markov models [4]. Such segmentation algorithms give high accuracy, but also require large amount of calculation resources. In some cases this is not possible, such as mobile devices, when calculation power is weak and/or network speed is limited. In such situations, it is needed to have an algorithm to extract segments as accurate as possible and to send only them through network for external processing. Common approach to speech signal zone identification is using a threshold value.

Threshold based segmentation works in this way: feature samples which exceed chosen threshold TH are marked as useful signal areas, see Fig. 1. In this case, if threshold is too low TH_{low} the various noisy segments will be marked as signal, if too high TH_{high} – important information at the beginning and the end may be lost. If it is known, that in the signal there is only one segment, it is possible to calculate threshold by evaluating noise samples in the beginning and the end of speech signal [5]. This algorithm for continuous speech will not work if there is no enough noisy signal at the ends of the signal. To have more accurate results, Lu proposed to use multi-feature segmentation supported by rule base to discriminate speech from music [6].

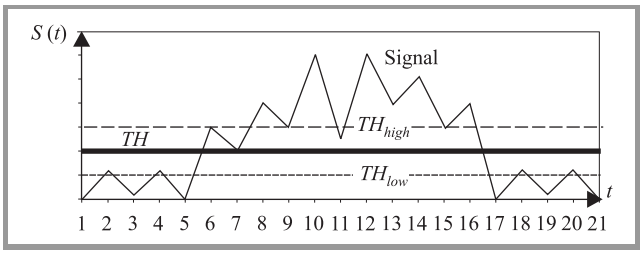

Fig. 1. Threshold-based segmentation.

To improve segmentation it is possible to use background knowledge of vocal tract and peculiarities of the language. Common segmentation errors: short peaks in the signal are noise segments, two segments with short space in between can be a sibilant consonant. This knowledge can be defined as a rule base, and be used in postprocessing of initial segmentation results. Rule based postprocessing of segments can give more stable results [7]. Off-line and on-line rules are working with different types of signals. Signals retrieved from corpus is possible to postprocess with off-line rule base. Such rules will not perform good in on-line mode, when the signal being processed from microphone.

1.1. Off-Line Rules

The postprocessing using a rule base can fix errors like segment interruption at the ends and short segments of noise. Waheed [7] proposed to use two rules (l_i – the ith segment length; $d_{i\,i+1}$ – distance between i and $i+1$ segment):

- if $l_i < minLength$ and $d_{i\,i+1} > minSpace$, then the segment i is discarded, similarly if $l_{i+1} < minLength$ and $d_{i\,i+1} > minSpace$, then the segment $i+1$ is discarded,
- if (l_i or l_{i+1}) $> minLength$ and $d_{i\,i+1} > minSpace$ and $(l_i + l_{i+1}) < maxLength$, then the two segments are merged, and anything between the two segments that was previously left, is made part of the speech.

The rule model proposed by Waheed works in off-line mode, when an earlier recorded signal is processed. This model for postprocessing requires to have two segments, that causes processing latency, $l_i + d_{i\,i+1} + l_{i+1}$, which can be not acceptable for automated speech recognition. On-line rules must cause smaller latency.

1.2. On-Line Rules

In on-line mode, when signal processing is part of a recording stream, a segment can be recognized faster if the recognition is done in parallel, as soon as this segment starts. For that purpose it is needed to have rules for current frame [8], see Fig. 2. Such rule base has validation latency. *minLength*, until recognition module can start to process the segment in parallel and *minSpace* – till the end of segment is found.

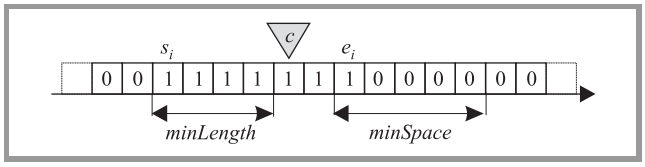

Fig. 2. Postprocessing rules for each frame: 0 – noise frame, 1 – speech frame, c – current position, s_i – segment start, e_i – segment end.

The rule base engine consists of working memory, assertions and a set of IF-THEN rules. The rule base inference is defined as a forward chaining system. The interpreter walks through the rules and applies them in order to take certain action. The rule is selected using a "first applicable" conflict resolution strategy, where rules have a specified order. Thus firing the rule with highest priority that matches current frame facts.
Rules specify how to act on the assertion set:

- **R1** $signalFrame \wedge noiseState$: mark frame as segment start,
- **R2** $signalFrame \wedge startState \wedge validLength$: accept start marker,
- **R3** $signalFrame \wedge endState$: join to previous segment,
- **R4** $noiseFrame \wedge startState \wedge \neg validSpace$: reject segment,
- **R5** $noiseFrame \wedge segmentState$: mark frame as segment end,
- **R6** $noiseFrame \wedge endState \wedge validSpace \wedge \neg validLength$: reject segment,
- **R7** $noiseFrame \wedge segmentState$: accept marked segment end.

The noise states can last as long as noise frames are processed. Same thing is applied for signal state with segment frames. From the start state the machine can go to a noise state if segment start has rejected (R6) or to a segment state if it approved (R2). From the end state it can go to noise state when segment end is approved (R7) or to segment state if segments are joined (R3).
On-line and off-line rule-based approaches are dependent on result that gives a segmentation algorithm. Threshold algorithms can be used for such frame classification, but this approach has weaknesses that were mentioned before. Automated syllables-like strong segments extraction was described by [9]. Similar extrema-based segmentation can be used to find strong elements in the signal. This paper presents such extrema-type rule-based algorithm.
This article is organized in four sections: Section 2 describes proposed segmentation algorithm, in Section 3 experiment results are presented and Section 4 is a conclusion.

2. Extrema-Based Segmentation with On-Line Rule-Based Processing

Proposed segmentation is based on detecting local minima and maxima of signal feature. Segments are constructed using extrema and processed with the help of a rule base (see Fig. 3).

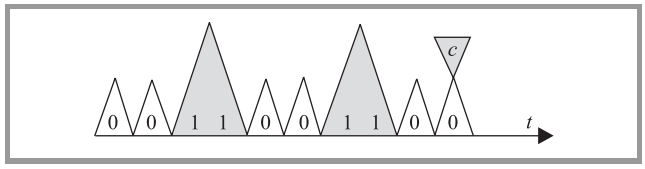

Fig. 3. Extrema-based segmentation: 0 – noise segment, 1 – speech segment, c – current position.

2.1. Extrema-Based Segmentation

Extrema-based segmentation is working with energy-like signal features. First of all a feature value is calculated for given signal sample. Local minimum and maximum are calculated. Atomic segments are initialized. A single atomic segment contains two minima and a single maximum in between. In the next step atomic segments are processed with a rule-based module and complex segments are constructed. A complex segment has its own features, as: A – area(power), S – number of sub-segments, L – length (see Fig. 4). Complex segments are classified into classes

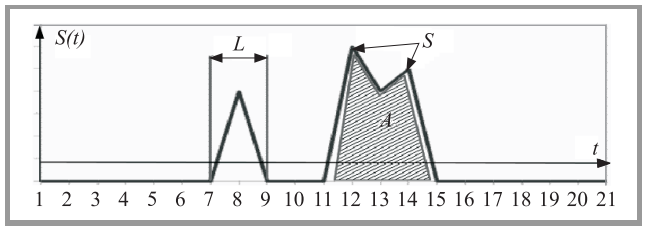

Fig. 4. Segment features.

using such defined segment features. Most powerful segments represent parts of vowels. Less powerful segments represent transitions between vowels (consonants) and others segments are background noises.

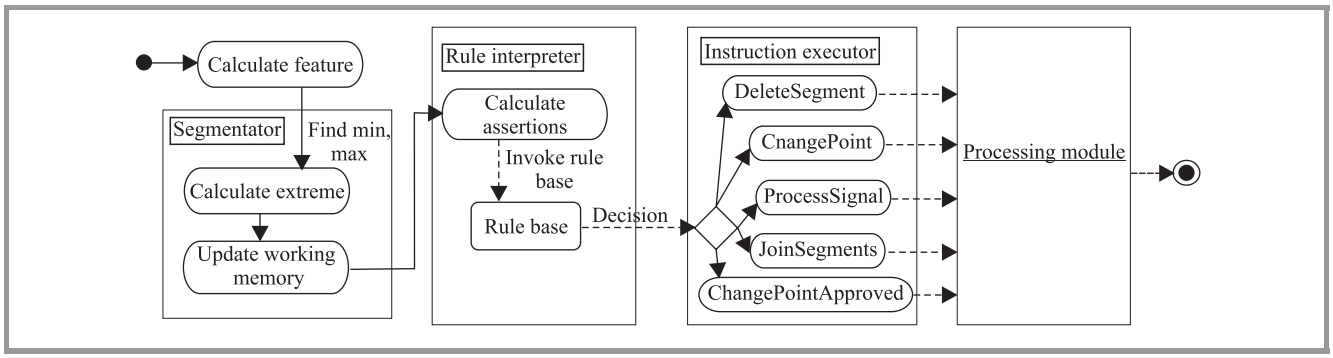

Fig. 5. Extrema-based segmentation with rule-based postprocessing.

2.2. On-Line Rule-Based for Extrema Segmentation

The segmentation algorithm state diagram is depicted in Fig. 5. The rule base receives calculated values from the segmentation module and it calculates assertions. Decision are made using these assertions in the rule interpreter module. Instruction executor send an event, dependent on the decision, to the processing module: automated speech recognition, automated speech corpus collector etc.
A simplified signal feature model shows how rules can be used, see Fig. 6. The processed signal has a lot of atomic segments. Extracted atomic segments has to be rejected or grouped into complex segments that point the areas where speech signal exists.

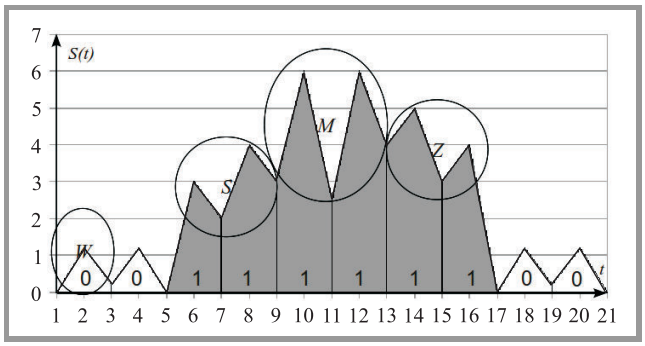

Fig. 6. Feature simplified segment model.

It is possible to define rules that can be used for segment union and rejection. Examples of possible rules: segments labeled by $S(signalIncreasing)$, $Z(signalDecreasing)$ and $M(variation)$ can be joined and $W(weakSegment)$ can removed.
Rules specify how to act on the assertion set:

- R1 $isMinimum$: a change point detected,
- R2 $isMaximum \wedge signalIncreasing$: join previous and current segments,
- R3 $isMaximum \wedge signalDecreasing$: join previous and current segments,
- R4 $isMaximum \wedge previousWeakSegment$: reject previous segments,
- R5 $isMaximum$: accept previous segment.

Rules interpreter is responsible for invoking certain actions and context changes, see Fig. 5. Such rules cause segment detection latency up to time point when a next maximum is found.
Proposed algorithm should perform better than a threshold algorithm in different environment types. The extrema-based algorithm should adapt automatically to different noises. It should work well with different features such as spectral flux, signal entropy, loudness, envelope, LPC residual. Experiments show [8] that rules-based segmentation results can be improved by using not a single feature, but several features in parallel.

3. Experiment

For the experiments there were compared 3 types of segmentation algorithms: threshold, dynamic (adaptive) threshold and the extrema rule-based algorithm. Threshold-based segmentation calculates the global statistics (histogram) of a feature for a complete recording and then determines a fixed decision threshold [10]. This algorithm showed good performance for single word detection in a signal. Dynamic threshold-based algorithm was working on same principle, only threshold was adjusted every 10 frames. Rule based segment extractor was working on proposed algorithm. Segmentation results of all 3 algorithms were postprocessed by the rule base [8] to make the results more accurate.
In experiment the noisy speech corpus Noizeus was used. It was developed to facilitate comparison of speech enhancement algorithms among research groups [11]. The noisy database contains 30 sentences (from three male and three female speakers), corrupted by 8 different real-world noises and with different SNRs. The noise signals were taken from the AURORA database: suburban train noise, babble, car, exhibition hall, restaurant, street, airport and train-station noise.
From the Noizeus speech corpus following speech samples were taken: “The birch canoe slid on the smooth planks.”, “He knew the skill of the great young actress.”, “Her purse was full of useless trash.”, “Read verse out loud for pleasure”, “Men strive but seldom get rich.”, “The sky that morning was clear and bright blue.”, “The set of china hit

the floor with a crash.". The Noizeus corpus distributes audio signal in 8 kHz 16 bit mono audio format. The noise signals were added to the speech signals at SNRs of 0 dB, 5 dB, 10 dB, and 15 dB. 198 speech samples were taken in total.

Every segmentation algorithm was applied to signals decomposed into 30 ms frames and with 66% overlap. Hamming window was used to minimize the speech signal discontinuities at the beginning and end of each analysis frame. First order infinite impulse response filter was used for pre-emphasis [5]. Automatically extracted segments were compared with marked by the expert segments.

3.1. Result Evaluation

Experiment results are evaluated with modified voice activity detector minimum performance standard TIA/EIA-136-250 [12]. In this standard there are 3 types of frames:

- Onset – few frames of speech at the beginning of speech segment.
- Steady – speech frames between onsets and offsets.
- Offset – few frames of speech at the end of speech segment.

Performance metrics are used:

- Probability of clipping speech onsets.
- Probability of detecting steady-state speech.
- Probability of clipping speech offsets.
- Normalized difference voice activity factor from truth.

These 4 error values are combined into one criterion. Ideal and testing segmentation results are processed in the same time frame by frame. When an ideal segment is started, the testing segment with the nearest boundaries are compared, see Fig. 7. Also delta voice activity factor is estimated in parallel.

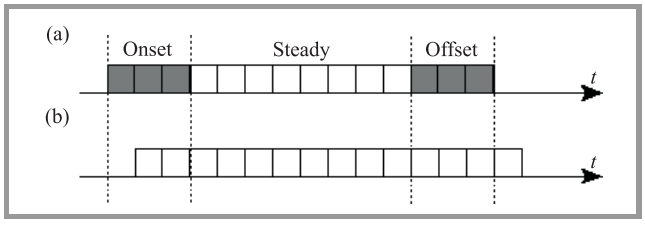

Fig. 7. Segment evaluation by TIA/EIA-136-250: (a) ideal signal, (b) testing signal.

3.2. Experiment Results

There were executed two types of experiments. The first one is to prove that rule based segmentation performs better than threshold segmentation. In the second it is tested if proposed algorithm shows better results in multi-feature segmentation.

3.3. Single Feature Segmentation

In the first experiment series the spectral flux feature was used as main segmentation feature. Spectral flux shows rate of spectral change in a signal. It was chosen as it showed good performance in speech segmentation [8]. The experimental results are shown in Fig. 8.

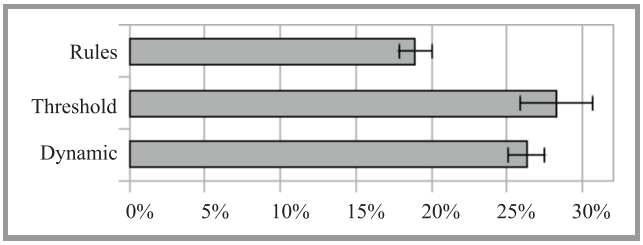

Fig. 8. Segmentation errors in results of threshold, dynamic threshold and rule-based segmentation.

Proposed rule algorithm overall performed better by 7.36% in comparison with dynamic threshold and 9.33% in comparison with static threshold. As expected dynamic threshold should perform slightly better than static one.

By comparing results in different type of noise types, extrema type rule base segmentation performed with smallest error ratio in all tested noise environments except the restaurant noise (see Fig. 9). In the restaurant environment

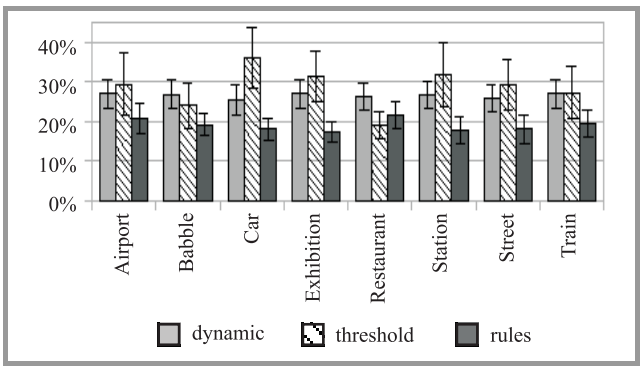

Fig. 9. Segmentation error ratio comparison by noise type.

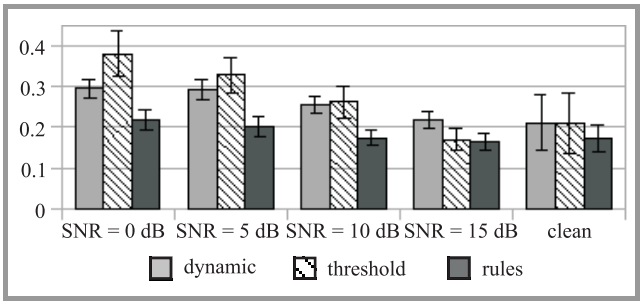

Fig. 10. Error ratio in rule, threshold and dynamic threshold-based segmentation when SNR rises.

Table 1
Error ratios indexed by SNR and noise types

Segmentation	SNR [dB]	Airport	Babble	Car	Exhibition	Restaurant	Station	Street	Train
Dynamic	0	29.93%	30.75%	29.46%	29.78%	29.31%	28.77%	29.31%	28.79%
	5	28.94%	29.57%	30.13%	28.65%	28.63%	28.46%	28.32%	30.69%
	10	26.73%	24.51%	22.59%	26.59%	26.29%	24.66%	24.94%	26.98%
	15	21.95%	22.32%	19.53%	23.18%	20.39%	24.74%	20.18%	21.13%
Threshold	0	47.57%	32.18%	55.54%	40.36%	17.38%	47.33%	28.65%	34.64%
	5	25.63%	26.10%	44.75%	37.83%	22.60%	42.29%	30.32%	32.41%
	10	30.42%	25.12%	25.88%	29.80%	16.10%	24.33%	32.38%	25.53%
	15	13.97%	12.59%	18.19%	17.00%	19.55%	13.37%	25.50%	16.03%
Rules	0	23.36%	23.64%	17.50%	20.10%	26.80%	20.19%	19.03%	23.02%
	5	23.61%	16.66%	20.94%	17.87%	24.92%	17.89%	19.23%	20.79%
	10	18.56%	19.25%	18.59%	14.78%	16.94%	14.97%	18.50%	18.05%
	15	17.16%	17.12%	15.52%	16.05%	17.24%	17.80%	15.58%	15.53%

music is played in background. The rule base made mistakes as this noise has many fluctuations. Proposed rules failed to merge and reject segments that normally were expected so. Future investigation of rule base enhancement may fix this problem. In the second place street and car noises leaded to most mistakes. Dynamic threshold method had stable results in all tested environments.

Figure 10 presents error ratio of segmentation results for different noise level. Extrema type rule-based segmentation is more stable in higher noise levels, than threshold-based segmentation. However dynamic threshold-based segmentation performed similar in SNR = 15 dB, this was caused by friendly conditions for noise level estimation. Rule-based showed better performance in SNR = 5 dB and SNR = 0 dB, as fluctuations created smaller segments and allows more accurate segment detection.

In Table 1 error ratios are listed by noise levels and types for each segmentation algorithm. Threshold segmentation showed the worst result of all segmentation algorithms in car noise with SNR = 0 dB, equals to 55.54% and the best result in babble with SNR = 15 dB, equals to 12.59%. The rule-based segmentation achieved the best result in exhibition noise, SNR = 10 dB equals to 14.78% and the worst result in restaurant noise, SNR = 0 dB equals to 26.80%. Threshold segmentation is less stable in different noise types, but can show good performance in non-nosy environments.

The processing speed of each segmentation algorithm was measured to find out how much CPU time extrema segmentation process requires. In average a signal of length 2.3 s was processed in: 50 ms threshold, 55 ms dynamic and 69 ms extrema. Threshold-based and rule-based segmentation time differ by 19 ms. This shows that increased power demand is not significant. Such an algorithm can be used in devices that have limited computation power.

3.4. Multi-Feature Segmentation

The aim experiment of the multi-feature segmentation experiment was to find out if parallel feature calculation can perform better that single feature.

For the second experiment 6 features were chosen: spectral flux, loudness, LPC residual, signal entropy, energy, envelope. These features showed good result in speech segmentation [8]. 41 possible combinations were created with 1 feature (6 combinations), 2 features (15 combinations) and 3 features (20 combinations). Every feature of each combination was processed in 3 steps:

- calculated feature values,
- segmented by selected algorithm,
- segmentation results were merged.

Merged final segmentation result was used to compare with expert segmentation.

Multi-feature segmentation results are presented in Fig. 11. It can be noted that the best result was showed by spectral flux and loudness features. Other feature combinations with spectral flux are in the top of the best results. spectral flux feature alone is in the 6th place. This experiment showed, that multiple feature usage may improve extrema type rule-based segmentation results. Although, it is needed deeper to investigate which feature group works better in noisy environments.

Feature combination selection for threshold-based segmentation was studied previously [8]. The order of feature combinations presented in Fig. 11 is similar to the experimental results with threshold segmentation. Although, in our experiment continuous speech signals were tested instead of short word commands, like in [8], also more types of noises were tested. It was expected that segmentation error ratio will increase for continuous speech segmentation.

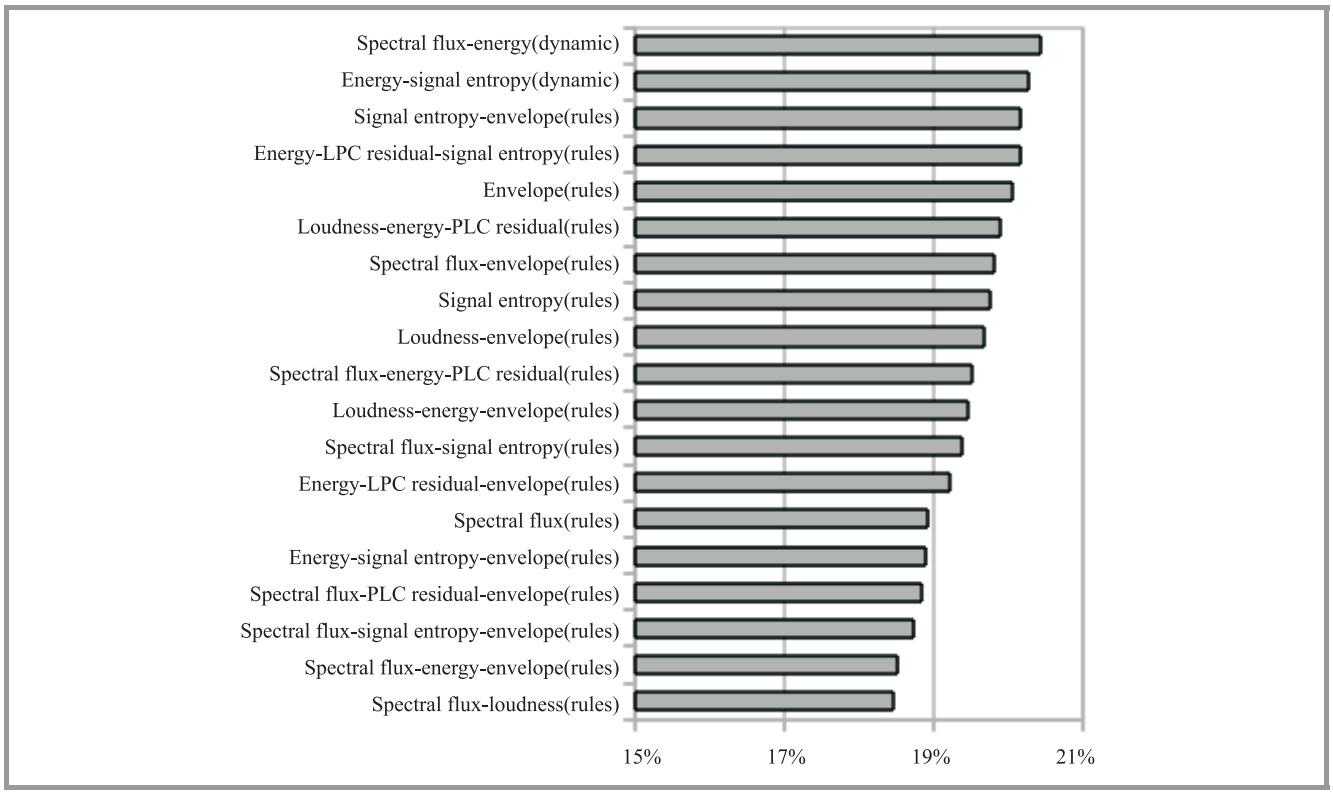

Fig. 11. Multi-feature segmentation results with rule, threshold and dynamic threshold algorithms.

The segmentation with features: spectral flux and envelope was higher error ratio in this experiment by 7.97% as in [8] experiment.

4. Conclusions

In this paper an extrema type rule-based segmentation algorithm was proposed:

- The main novelty is the postprocessing of extrema-based segments by using several rules of segment merging and deleting.
- The experiments showed that extrema rule-based segmentation performed better than threshold or dynamic threshold-based algorithms by around 7%.
- The rule-based approach showed better result in high noise levels environments.
- Multi-feature segmentation experiment showed that the proposed algorithm it may also improve results when using 2 or 3 features in parallel.

The rule-based algorithm showed weak results in some noise type environments, this will be improved by tuning rules in future works.

References

[1] Y. Hioka and N. Hamada, "Voice activity detection with array signal processing in the wavelet domain", *IEICE Trans. Fund. Elec. Commun. Comput. Sci.*, vol. 86, no. 11, pp. 2802–2811, 2003.

[2] F. Beritelli and S. Casale, "Robust voiced/unvoiced speech classification using fuzzy rules", in *IEEE Worksh. Speech Cod. Telecommun. Proc.*, Pocono Manor, USA, 1997, pp. 5–6.

[3] Y. Qi and B. R. Hunt, "Voiced-unvoiced-silence classifications of speech using hybridfeatures and a network classifier", *IEEE Trans. Speech Audio Proces.*, vol. 1, no. 2, pp. 250–255, 1993.

[4] S. Basu, "A linked-HMM model for robust voicing and speech detection", in *IEEE Int. Conf. ICASSP'03*, Hong Kong, China, 2003, vol. 1, pp. 816–819.

[5] J. Lipeika, A. Lipeikiene, and L. Telksnys, "Development of isolated word speech recognition system", *Informatica*, vol. 13, no. 1, pp. 37–46, 2002.

[6] L. Lu, H. Jiang and H. J. Zhang, "A robust audio classification and segmentation method", in *Proc. 9th ACM Int. Conf. Multimedia*, Ottawa, Canada, 2001, p. 211.

[7] K. Waheed, K. Weaver, and F. Salam, "A robust algorithm for detecting speech segments using an entropic contrast" in *Proc. IEEE Midwest Symp. Circ. Sys. MWCAS 2002*, Tulsa, USA, 2002, vol. 5.

[8] M. Greibus and L. Telksnys, "Speech segmentation features selection", *Inf. Technol.*, vol. 15, no. 4, pp. 33–45, 2009.

[9] P. Mermelstein, "Automatic segmentation of speech into syllabic units", *J. Acoust. Soc. Am*, vol. 58, no. 4, pp. 880–883, 1975.

[10] S. Van Gerven and F. Xie, "A comparative study of speech detection methods", in *Proc. 5th Eur. Conf. EUROSPEECH'97*, Rhodes, Greece, 1997, vol. 3, Rhodes, Greece pp. 1095–1098.

[11] Y. Hu and P. C. Loizou, "Subjective comparison and evaluation of speech enhancement algorithms", *Speech Commun.*, vol. 49, no. 7–8, pp. 588–601, 2007.

[12] *TIA/EIA-136-250 Standard*, "TDMA third generation wireless – minimum performance standards for acelp voice activity detection. Tech. Rep., TIA, 2001.

Mindaugas Greibus was born in Kaunas, Lithuania, in 1979. In 2005 he received the M.Sc. from the Vytautas Magnus University, Kaunas, Lithuania. He currently is pursuing a Ph.D. at the Institute of Mathematics and Informatics.

e-mail: mindaugas.greibus@exigenservices.com
Institute of Mathematics and Informatics
Recognition Processes Department
Gostauto 12
LT-01108 Vilnius, Lithuania

Laimutis Telksnys is Professor, Doctor Habilitatis in informatics, Doctor Honoris Causa of the Kaunas University of Technology, member of Lithuanian Academy of Sciences, head of Recognition Processes Department at the Institute of Mathematics and Informatics, Vilnius, Lithuania. He is the author of an original theory of detecting changes in random processes, investigator and developer of a computerized system for statistical analysis and recognition of random signals. His current research interests are in analysis and recognition of random processes, cardiovascular signals and speech processing and computer networking.
e-mail:
Institute of Mathematics and Informatics
Recognition Processes Department
Gostauto 12
LT-01108 Vilnius, Lithuania